Koel-TTS: Enhancing LLM based Speech Generation with Preference Alignment and Classifier Free Guidance

Shehzeen Samarah Hussain, **Paarth Neekhara**, **Xuesong Yang**, **Edresson Casanova**
Subhankar Ghosh, **Mikyas T. Desta**, **Roy Fejgin**, **Rafael Valle**, **Jason Li**
NVIDIA Corporation, USA

Abstract

Autoregressive speech token generation models produce speech with remarkable variety and naturalness but often suffer from hallucinations and undesired vocalizations that do not conform to conditioning inputs. To address these challenges, we introduce Koel-TTS, an encoder-decoder transformer model for multilingual TTS that improves contextual adherence of speech generation LLMs through preference alignment and classifier-free guidance (CFG). For preference alignment, we design a reward system that ranks model outputs using automatic metrics derived from speech recognition and speaker verification models, encouraging generations that better match the input text and speaker identity. CFG further allows fine-grained control over the influence of conditioning inputs during inference by interpolating conditional and unconditional logits. Notably, applying CFG to a preference-aligned model yields additional gains in transcription accuracy and speaker similarity, demonstrating the complementary benefits of both techniques. Koel-TTS achieves state-of-the-art results in zero-shot TTS, outperforming prior LLM-based models on intelligibility, speaker similarity, and naturalness, despite being trained on significantly less data. [1]

1 Introduction

The advancement of large language models (LLMs) has brought transformative improvements to speech synthesis, enabling more natural and contextually adaptive speech generation. In particular, there has been a recent surge in the use of LLMs for various applications such as text-to-speech (TTS) and speech-to-speech translation (Wang et al., 2023; Zhang et al., 2023; Borsos et al., 2023; Neekhara et al., 2024a; Yang et al., 2024; Susladkar et al., 2024; Wang et al., 2024). LLM-based TTS systems enable prompt-based customization, generating speech with human-like intonation while adapting to stylistic cues, contexts, and expressive nuances. This allows for diverse applications, from conversational interfaces to expressive narration, without extensive retraining. However, LLM-based TTS systems face challenges, with hallucinations being a prominent issue (Sahoo et al., 2024; Song et al., 2024; Neekhara et al., 2024a; Borsos et al., 2023). For example, when encountering text with repeated or redundant phrases, LLM-based TTS models may overemphasize these repetitions or fail to capture the intended flow and naturalness of the sentence. Additionally, among the multiple outputs sampled for the same input, there can be significant variation in quality, with some outputs sounding more natural, accurate, and appealing than others. This issue is akin to challenges faced in text-generation LLMs, where outputs may range from highly coherent to erroneous, depending on the model's response to complex prompts.

To tackle these challenges, we propose preference alignment and CFG techniques to enhance contextual coherence of LLM-based TTS models. We introduce Koel-TTS, a transformer-based autoregressive TTS model that leverages a low-frame-rate (21.5 FPS) audio codec (Casanova et al., 2025) to enable low-latency speech generation. To perform preference alignment, we first identify key metrics that strongly correlate with human judgments of generated speech: transcription accuracy and target speaker similarity. Each metric captures distinct aspects of the generated output and can be evaluated using automatic speech recognition (ASR) and speaker verification (SV) models. We integrate these metrics into a reward system that ranks the generated outputs. With this foundation, we then explore preference alignment algorithms, focusing on pairwise ranking methods and scalar reward optimization. Our findings show that fine-tuning the base model with preference alignment significantly improves speaker similarity, intelligi-

[1] Audio Examples: `https://koeltts.github.io/`

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21219–21234
November 4-9, 2025 ©2025 Association for Computational Linguistics

bility, and generalization to unseen speakers. More interestingly, our method also enhances naturalness, despite not explicitly optimizing for this metric.

To further enhance adherence to conditioning inputs using CFG, we train the Koel-TTS model with both conditional and unconditional inputs by randomly dropping out the text and context audio during training. At inference time, we interpolate between the unconditional and conditional logits using a CFG scale, resulting in notable improvements in intelligibility, speaker similarity, and naturalness. Importantly, CFG can be applied independently to either the base model or the preference-aligned model, consistently boosting performance across all evaluation metrics. By combining preference alignment with CFG, we train a 1.1 billion parameter multilingual Koel-TTS model that achieves state-of-the-art zero-shot TTS performance across several human and automatic evaluations. The key contributions of this work are as follows:

- We introduce Koel-TTS, a multilingual encoder-decoder transformer model that maps text and context audio directly to acoustic tokens using a low-frame-rate audio codec, enabling expressive, robust, and low-latency autoregressive speech synthesis.
- We propose a preference alignment framework for LLM-based TTS using ASR and SV models as reward signals, showing that carefully curated text-context pairs and principled ranking approach to favor high-quality generations, are key to maximizing alignment gains.
- We adapt CFG for LLM-based speech synthesis, which involves dropping out both the text and context audio conditioning during training. We demonstrate that CFG can significantly improve naturalness, speaker similarity, and intelligibility even for token-based speech LLMs trained with next token prediction loss.
- Our Koel-TTS model, trained with preference alignment and CFG, achieves state-of-the-art zero-shot TTS performance while reducing hallucinations and improving intelligibility. Our model implementation is publicly available in the Koel-TTS repository[2].

2 Methodology

Koel-TTS is an autoregressive speech token generation model conditioned on a text transcript and an audio prompt from the speaker. We begin by outlining the tokenization scheme and model architecture, and then introduce two key techniques — preference optimization and CFG to enhance the model's robustness and speaker similarity.

2.1 Tokenization

Speech: We employ a neural audio codec model to transform raw speech signals into tokenized representations. For a given audio signal $\mathbf{a}$, the codec model outputs a two-dimensional acoustic matrix $\mathbf{C}_{T\times N} = \mathit{CodecModel}(\mathbf{a})$. In this representation, $\mathbf{C}_{T\times N}$ consists of m-bit discrete codes, where T corresponds to the downsampled sequence length, and N represents the number of codebooks per timestep. We use the Low Frame-rate Speech Codec (Casanova et al., 2025), which achieves high-quality audio compression at a bitrate of 1.89 kbps and a frame rate of 21.5 frames per second, utilizing N=8 independent codebooks. The codec uses Finite Scalar Quantization (FSQ) (Mentzer et al., 2024), which ensures independence among the codebooks. This independence eliminates the need for additional models or delay mechanisms, enabling the parallel prediction of all N codebooks at each timestep.

Text: We explore two text tokenization methods: phonemes and characters. Phonemes, commonly used in neural TTS, capture fundamental sound units but require language-specific grapheme-to-phoneme (G2P) conversion. In contrast, character-based tokenization eliminates this need, enabling direct conversion of graphemes to acoustic information. In our experiments, we use IPA phonemes and character tokenizers for English, German, and Spanish, while applying only character tokenizers for other languages. We utilize an aggregated tokenizer that maintains separate token embeddings for each language. Additionally, we perform an ablation study with a shared character tokenizer and a multilingual sentencepiece tokenizer across all languages, with results detailed in Appendix E.

2.2 Model Architecture

Our speech generation model is an autoregressive (AR) transformer decoder conditioned on text encodings from a non-autoregressive (NAR) transformer encoder using cross-attention (Figure 1).

[2] Koel-TTS model has been renamed to MagpieTTS in the repository: `https://github.com/NVIDIA-NeMo/NeMo/tree/magpietts_2503`

This encoder-decoder architecture allows us to encourage monotonic text and speech alignment through an attention prior and CTC loss on cross-attention scores without interfering with the self-attention mechanism of the AR decoder (Section 2.3). The AR transformer predicts audio tokens frame by frame, generating all N codebooks in parallel at each time step, conditioned on previous predictions and the cross-attention inputs. At each decoder timestep, the input acoustic embedding is derived by referencing and summing the embedding of each of the N codebooks.

To enable speaker and style conditioning through context audio (alternate audio from the target speaker), the context audio tokens are directly prepended to the target audio tokens. In addition to the above described decoder-context mechanism for speaker conditioning, we explore two additional architectures—SV Conditioned Koel-TTS and Multi-Encoder Koel-TTS—in Appendix A.

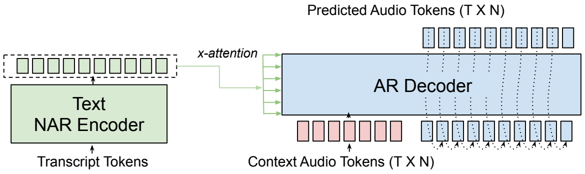


Figure 1: Koel-TTS Decoder Context Model Architecture

2.3 Training Objective

The output of each decoder-timestep is mapped to a vector of size $N \times 2^m$ using a linear layer to obtain the logits of all N codebooks (each of size m-bits) at that timestep. Thereby, for all decoder time-steps, we obtain logits ℓ of size $T \times N \times 2^m$ and calculate cross-entropy as follows:

$$\mathcal{L}_{token} = CE\left(softmax\left(\ell\right), target_{N \times T}\right)$$

In addition to the above, to improve text and speech alignment, past work (Neekhara et al., 2024a) recommends biasing the cross-attention scores between the transcript encoder and AR decoder to be monotonic, using an attention prior and Connectionist Temporal Classification (CTC) loss. Specifically, given the cross-attention score matrix $\mathbf{A}^{l,h}_{T \times M}$, of the h^{th} cross-attention head in decoder layer l, between the audio timesteps (T) and text timesteps (M), we generate a static prior using the 2D beta-binomial distribution $\mathbf{P}_{T \times M}$. Given this prior, we obtain the re-scaled attention scores as:

$$\mathbf{A}^{l,h}_{T \times M} \leftarrow \mathbf{A}^{l,h}_{T \times M} \odot \mathbf{P}_{T \times M}$$

The attention prior is applied for the first $10{,}000$ training iterations and then linearly annealed to a uniform distribution (all ones) for the next $5{,}000$ iterations and turned off thereafter. Turning off the prior is necessary since we cannot use this prior during inference[3] and annealing ensures stability during training.

Additionally, to encourage valid monotonic sampling from the alignment matrix, we calculate likelihood of all possible monotonic reductions using the CTC algorithm. That is, given the alignment matrix $\mathbf{A}^{soft_{l,h}}_{T \times M} = softmax(\mathbf{A}^{l,h}_{T \times M})$, we obtain the alignment loss for a decoder layer and head as:

$$\mathcal{L}^{l,h}_{align} = CTCLoss\left(\mathbf{A}^{soft_{l,h}}_{T \times M}, q_M\right)$$

where $q_M = \{1, 2, \ldots M\}$ is the target monotonic sequence from 1 to M. The alignment loss is summed across all cross-attention heads and layers to obtain $\mathcal{L}_{align} = \sum_{l,h} \mathcal{L}^{l,h}_{align}$. The final training loss is obtained as $\mathcal{L} = \mathcal{L}_{token} + \alpha \mathcal{L}_{align}$, where α is the alignment loss coefficient set as 0.002.

2.4 Preference Alignment

We employ preference optimization methods to steer the outputs of Koel-TTS towards more desirable results. For a given text and context audio input $x = (x_{text}, x_{audio})$, the model's response distribution $\pi(y|x)$ encompasses a range of potential outputs y with varying levels of alignment to the desired criteria. By constructing a dataset that explicitly labels certain responses y_c as chosen and others y_l as rejected, we can leverage preference-based optimization algorithms to shift the model's distribution toward producing more preferred responses.

One such approach is Direct Preference Optimization (DPO) (Rafailov et al., 2024). DPO uses preference comparisons to modify the policy π by contrasting it against a reference policy π_{ref}. Specifically, given an input x and a chosen response y_c that is preferred over a rejected response y_l, DPO seeks to increase the likelihood ratio $\frac{\pi(y_c|x)}{\pi_{\text{ref}}(y_c|x)}$ relative to $\frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$. The core objective can be expressed as:

$$\mathcal{L}_{\text{DPO}} = \mathbb{E}_{x,y_c,y_l}\left[\beta \log \frac{\pi(y_c|x)}{\pi_{\text{ref}}(y_c|x)} - \beta \log \frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right]$$

where β is a parameter for controlling the deviation from the base reference policy π_{ref}. The above

[3] The final audio sequence length is unknown during inference.

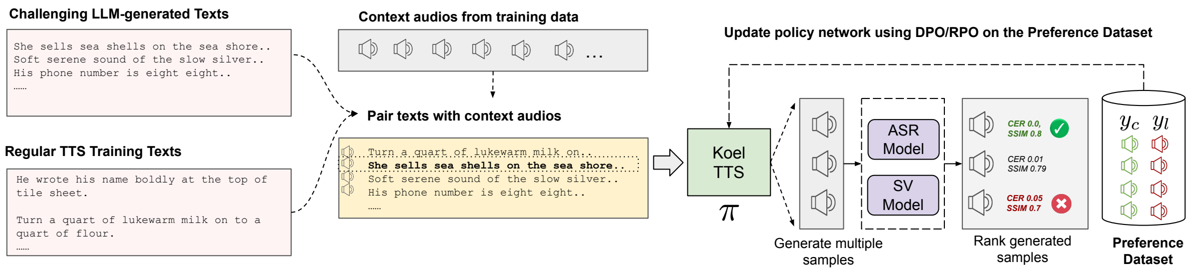

Challenging LLM-generated Texts
She sells sea shells on the sea shore..
Soft serene sound of the slow silver..
His phone number is eight eight..
......
Regular TTS Training Texts
He wrote his name boldly at the top of tile sheet.
Turn a quart of lukewarm milk on to a quart of flour.
......
Context audios from training data
...
Pair texts with context audios
Turn a quart of lukewarm milk on..
She sells sea shells on the sea shore..
Soft serene sound of the slow silver..
His phone number is eight eight..
......
Update policy network using DPO/RPO on the Preference Dataset
Koel TTS
π
ASR Model
SV Model
CER 0.0, SSIM 0.8
CER 0.01 SSIM 0.79
CER 0.05 SSIM 0.7
Generate multiple samples
Rank generated samples
y_c y_l
Preference Dataset

Figure 2: Preference Alignment for Koel-TTS: Koel-TTS generates multiple outputs for challenging text and context audio prompts, which are then rewarded using ASR and SV models to create a preference dataset for DPO and RPO.

formulation, encourages π to produce responses more similar to y_c than y_l, effectively aligning the model with the desired preferences.

Building upon DPO, we also leverage Reward-aware Preference Optimization (RPO) (Adler et al., 2024), which considers the magnitude of reward differences in the optimization process. Rather than treating all chosen versus rejected distinctions as equal, RPO utilizes scalar rewards to measure how much better the chosen response is compared to the rejected one. The RPO objective introduces a factor that scales the preference updates based on the reward gap $r^*(x, y_c) - r^*(x, y_l)$ as follows:

$$\mathcal{L}_{\text{RPO}}(x, y_c, y_l) = \mathbb{D}\left[\beta \log \frac{\pi(y_c \mid x)}{\pi_{\text{ref}}(y_c \mid x)} - \beta \log \frac{\pi(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right. \\ \left. \Big\| \; \eta\big(r^*(x, y_c) - r^*(x, y_l)\big)\right]$$

where η is a scaling parameter and $\mathbb{D}$ is a distance metric given by $\mathbb{D}\left[a \| b\right] := \sigma(b) \log \frac{\sigma(b)}{\sigma(a)} + (1 - \sigma(b)) \log \frac{1-\sigma(b)}{1-\sigma(a)}$ Thereby, RPO mitigates overfitting to narrowly better responses since the loss value is scaled as per the reward difference.

Preference Data Creation and Reward System: To construct the preference dataset (x, y_c, y_l), we begin by selecting a diverse set of text and speaker prompt combinations that challenge the model's ability to produce accurate and natural speech. The text data includes a mix of regular sentences from standard TTS datasets, and carefully curated challenging transcripts generated by prompting text LLMs. These challenging texts are designed to test the model's robustness and include elements such as repeated words, numbers, and phonetically complex sequences. The inclusion of standard texts ensures generalizability, while the challenging examples target specific weaknesses of the TTS model as illustrated in Figure 2.

For each text and speaker prompt, we generate P audio samples using multinomial sampling at temperature=`0.7`. Each generation is evaluated using the Parakeet TDT 1.1B ASR (Xu et al., 2023) and Titanet-large SV (Koluguri et al., 2022a) models. Specifically, we obtain the character error rate (CER) between the transcript of the generated audio and input text using the ASR model, and cosine similarity (SSIM) between the embeddings of the context audio and the generated audio obtained from the SV model. Based on the CER and SSIM, we perform Pareto optimal ranking (Deb, 2011) on the set of P generated audio samples for a given input pair—First, we identify the Pareto front, which consists of all audio samples that are not dominated by any other sample. That is, no other audio is strictly better on at least one metric and equally good or better on the other. Once we identify the first Pareto front, we remove those samples and repeat the process on the remaining audios to find the next front, and so on. Within each Pareto front, we prioritize samples by assigning higher ranks to those with lower CER scores. If there are ties based on CER, we further differentiate by favoring samples with higher SSIM values. We detail this pareto ranking procedure in Appendix D.

After ranking the examples, we select the highest-ranked as chosen and the lowest-ranked as rejected for DPO, since we empirically find high-contrast pairs to be beneficial for DPO. For RPO, which handles scalar reward differences, we pair the top two with the bottom two in all combinations. In both cases, we discard pairs where the chosen example scores worse on any metric (CER or SSIM) than the rejected one.

To assign scalar rewards for RPO, we normalize the CER and SSIM differences between the chosen and rejected examples, and set the reward gap as:

$$r^*(x, y_c) - r^*(x, y_l) = \Phi(\Delta\widetilde{\text{CER}}) + \Phi(\Delta\widetilde{\text{SSIM}})$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution, and $\Delta\widetilde{\text{CER}}$ and $\Delta\widetilde{\text{SSIM}}$ are the normalized differences of CER and SSIM respectively, between the chosen and rejected examples.

2.5 Classifier Free Guidance

To adapt CFG for autoregressive token prediction models, we train both a conditional and an unconditional model simultaneously by randomly dropping out the text and context/speaker conditioning during training. At inference time, conditional and unconditional outputs are combined to guide the speech generation process. This approach allows for more precise control over the generated speech, which can lead to improved pronunciation, prosody, robustness, and overall audio quality.

Distinct from the previous work that only deals with text-independent conditionals (Darefsky et al., 2024), in our approach, we randomly dropout both audio and text conditioning inputs (with a probability of 10%) during training and interpolate conditional logits (ℓ_c) with the unconditional logits (ℓ_u) during inference,

$$\ell_{cfg} = \gamma * \ell_c + (1 - \gamma) * \ell_u$$

where $\gamma \geq 1$ is the CFG interpolation scale controlling the strength of guidance. Higher scale values steer the generation to follow the text/audio inputs, while lower scale values allow more variations. In practice, we sweep around a range of values to find the optimal scale γ. Figure 3 demonstrates the CFG inference process.

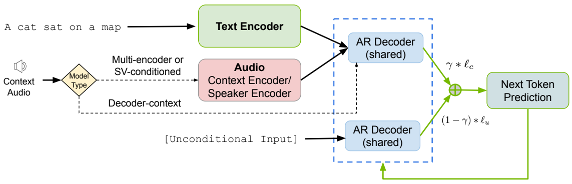


Figure 3: CFG Inference: Logits from conditional and unconditional inputs are combined using a CFG scale γ>1, steering model predictions towards better alignment with conditional inputs.

3 Experiment Setup

3.1 Datasets

For our primary experiments, we train the models on a datablend containing $18k$ hours of English TTS data from the following datasets: *train-clean-360* and *train-clean-100* subsets of LibriTTS (Zen et al., 2019), HiFiTTS (Bakhturina et al., 2021), a $17k$-hour subset of the LibriVox MLS dataset (Pratap et al., 2020) and a proprietary, 2-speaker, 63-hour dataset.

For multilingual TTS, we investigate six languages English, Spanish, German, French, Italian, and Dutch. For non-English languages, we use the CML dataset (Oliveira et al., 2023) that contains 1,562 hours of German, 642 hours of Dutch, 476 hours of Spanish, 283 hours of French, 131 hours of Italian speech data. Additionally, we incorporate 42 hours of internal Spanish data from two speakers. Combining this with our $18k$ hours of English TTS data, we create a final blend of $21k$ hours of multilingual TTS data.

3.2 Baseline Model Training

With the above datasets, we create *(context audio, transcript, target audio)* triplets where context and target audio are distinct utterances from the same speaker. During training, we use a random 5 second slice of the context audio. For our primary experiments, we train a 380 million parameter model (*Koel-TTS 380m English*) that consists of 12 decoder transformer layers using a hidden dimension of 768 and a feed-forward network (FFN) dimension of 3072. Rather than a standard FFN sub-layer, our decoder uses a causal convolution layer with a kernel size 3. The transcript encoder comprises of 6 transformer layers that do not use causal masking, but otherwise match the decoder's specifications. We use multi-headed self and cross attention layers with 12 heads in each layer. In addition to the above archiectute, we train a larger 1.1 billion parameter model for multilingual-TTS (*Koel-TTS 1.1b Multilingual*). In this architecture, we use 16 decoder layers and 6 encoder layers, with hidden dimension of 1536 and FFN dimension of 6144. Other hyper-parmaters match the *Koel-TTS 380m English* model.

The Koel-TTS 380m English model is trained on 16 NVIDIA A100 GPUs using a global batch size of 256, optimized using Adam optimizer with an initial learning rate of $1e-4$. The learning rate is annealed every 1000 training steps using an exponential decay factor of 0.998. Training on the English dataset converges in around $200k$ steps. The Koel-TTS 1.1b Multilingual follows the same training procedure and hyper-parameters on 32 NVIDIA A100 GPUs. Training for the 1.1b model converges in around $150k$ steps.

3.3 Preference dataset and alignment

To perform preference alignment, we create a preference dataset using the procedure described in Section 2.4. Specifically, we first curate 800 challenging texts generated using Llama-8b (Touvron et al., 2023). It is prompted to generate texts with repeated words and alliterations. The complete list of these texts can be found on our webpage. We pair each challenging text with 10 random context audios sampled from our training dataset. Next, we sample 50,000 regular transcripts from our training data, and pair each text with one random context audio from our training data. This results in a total 58,000 text and context audio pairs. For preference alignment of the multilingual model, we create $10k$ text and context audio pairs per language (by pairing texts with a random context audio), from the CML training data of each language. We combine these pairs with $20k$ English text and context audio pairs randomly sampled from the $58k$ pairs used in our primary experiments. We utilize the *whisper-large-v3* (Radford et al., 2022) ASR model in our reward system to create preference pairs.

For each pair, we generate 6 audio samples from Koel-TTS and create chosen-rejected pairs using the reward and filtering criteria outlined in Section 2.4. Starting from our base checkpoints, we perform DPO or RPO finetuning for a maximum of 4,000 mini-batch iterations using a batch-size of 64 pairs, optimized using Adam optimizer with a fixed learning rate (LR) `2e-7`. For RPO, we use β=`0.01` and η=`1.0`; for DPO we try β=`0.01` and β=`0.05` and choose the checkpoint with the best validation metrics.

3.4 Evaluation

We evaluate synthesized speech on intelligibility, speaker similarity, and naturalness. Intelligibility is measured using ASR-based character error rate (CER) and word error rate (WER), with Parakeet-TDT (Xu et al., 2023) for English and *whisper-large-v3* (Radford et al., 2022) for other languages. Speaker similarity is assessed via cosine similarity (SSIM) between speaker embeddings of synthesized and context audio, using *Titanet-Small* (Koluguri et al., 2022a) which is different from the *Titanet-Large* used for preference alignment. Naturalness is evaluated with Squim-MOS (Kumar et al., 2023), and we also conduct a human evaluation for two of our zero-shot Koel-TTS models, benchmarking them against others in Section 4.1. For inference, we use multinomial sampling with temperature=`0.6`. Due to probabilistic generation, each experiment is repeated five times, reporting mean metrics with 95% confidence intervals.

For unseen English speakers, we create a subset of *test-clean* LibriTTS containing 180 utterances from a total of 36 out of the 40 speakers, using 5 distinct context and target audios from each speaker. We use a random 5 second slice from the context audio during inference for all experiments. For non-English languages, we use 100 speaker-balanced utterances for each language from the CML test set.

4 Results

We report the results of the Baseline model and the improvements through preference alignment and CFG for the Koel-TTS 380m English model in Table 1. As evident, both DPO and RPO significantly improve intelligibility and speaker-similarity metrics over the baseline. Interestingly, preference alignment significantly improves zero-shot speaker similarity on unseen speakers, even though we do not include any new speakers in the preference data creation. The naturalness metric Squim-MOS also shows an improvement over the baseline models, even though we don't explicitly include it in our reward system. This suggests that CER and SSIM metrics serve as good proxy for human preferences and can be automatically computed, thereby allowing easy scaling up of the preference alignment process. In practice, we find RPO to be less sensitive to hyper-parameter tuning and number of training iterations than DPO. RPO also works reliably when preference data does not have high-contrast chosen-rejected pairs, since it considers reward differences instead of a binary pair, in its optimization objective.

Table 1: Preference Alignment (DPO, RPO) and CFG improvements over a baseline *Koel-TTS 380m English* model for zero-shot TTS. Both methods improve intelligibility, speaker similarity and naturalness metrics, with best results achieved when they are used together.

Model/Technique	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS
Ground Truth	0.80	1.83	0.771	3.93 ± 0.03
Baseline (BL)	2.68 ± 1.13	4.02 ± 1.12	0.637 ± 0.008	4.35 ± 0.02
BL + DPO	0.89 ± 0.15	1.90 ± 0.28	0.667 ± 0.003	4.40 ± 0.01
BL + RPO	1.17 ± 0.94	2.09 ± 1.00	0.681 ± 0.005	4.40 ± 0.01
BL + CFG	0.57 ± 0.11	$\mathbf{1.37 \pm 0.11}$	0.720 ± 0.004	$\mathbf{4.42 \pm 0.01}$
BL + DPO + CFG	$\mathbf{0.55 \pm 0.10}$	1.42 ± 0.28	$\mathbf{0.729 \pm 0.003}$	4.41 ± 0.01
BL + RPO + CFG	$\mathbf{0.55 \pm 0.11}$	1.41 ± 0.19	$\mathbf{0.729 \pm 0.003}$	$\mathbf{4.42 \pm 0.01}$

For CFG, by controlling the scale γ during inference, we can steer the generations to be better aligned with conditional inputs. We vary γ between 1 to 3 at 0.2 intervals and show the results of this experiment in Figure 5. Increasing γ significantly reduces the CER and simultaneously increases SSIM across all models. From these observations, we

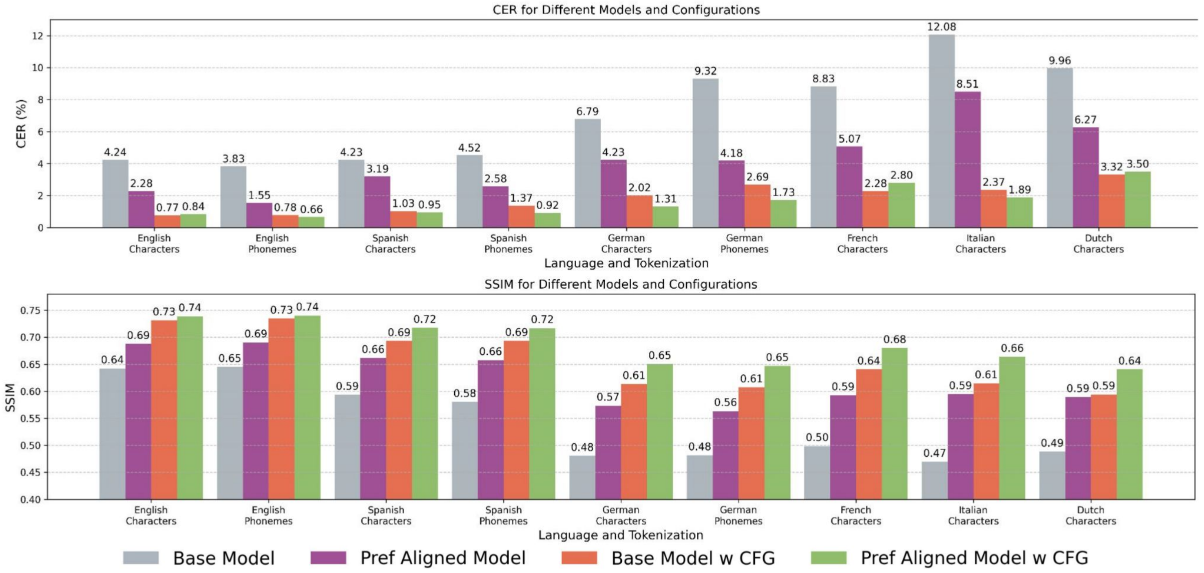


Figure 4: Evaluations for the *Koel-TTS 1.1b Multilingual model* across various languages and text tokenizers. Both CFG and preference alignment, independently and in combination (green), improve CER and SSIM over the base model (gray).

set γ=2.5 as the optimal value. Additionally, CFG inference on a preference aligned model results in further improvements across all metrics (Table 1). The reduction in CER/WER confidence intervals indicates that we can generate accurate speech reliably. In Appendix B, we report results for English-TTS on alternate model architectures and observe similar improvements from preference alignment and CFG across all experiments.

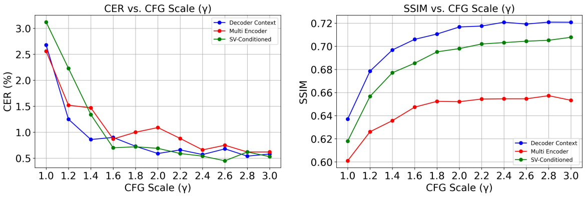


Figure 5: Effect of CFG scale on CER/SSIM for Koel-TTS 380m English Model

Figure 4 presents the results of multilingual-TTS evaluations on unseen speakers from each language. As shown by the results, both preference alignment and CFG (with γ=2.5) yield substantial improvement in both intelligibility and speaker similarity metrics across various languages and tokenizers. More interestingly, CFG inference on a DPO fine-tuned checkpoint, yields substantial speaker similarity improvements over using either DPO or CFG in isolation, especially for non-English languages.

We find that Koel-TTS can work effectively on raw character tokens, and achieve similar results as using phonetic inputs, for languages in which we consider both phoneme and character tokenizers (English, Spanish and German). Incorporating both CFG and DPO, *Koel-TTS 1.1b Multilingual* achieves similar CER as the *Koel-TTS 380m English* model and improves speaker similarity (0.740 vs. 0.726). We present ablations with alternate multilingual tokenization schemes in Appendix E.

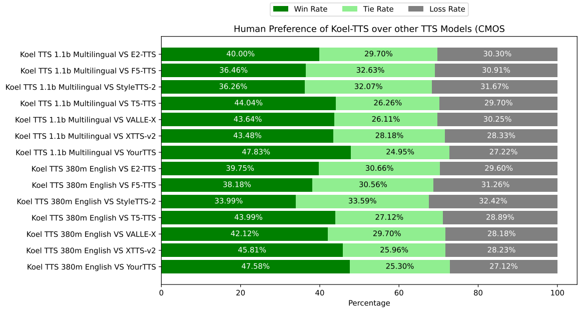


Figure 6: Koel-TTS vs. Previous Models: Dark green bars indicate the percentage of instances where human listeners preferred Koel-TTS for audio naturalness during side-by-side evaluations.

4.1 Comparison against Past Work

We benchmark both the *Koel-TTS 380m English* and *Koel-TTS 1.1b Multilingual* models against past work and open source models. We evaluate all models for zero-shot TTS on the unseen speaker evaluation set (test-clean LibriTTS subset), using the same evaluation procedure as described in Section 3.1. We also compute three human evaluation metrics on Amazon Mechanical Turk namely Naturalness Mean Opinion Score (MOS), Speaker similarity MOS (SMOS) and Comparative MOS (CMOS). For complete details on MOS studies, see Appendix H. As shown in Table 2, Koel-TTS achieves state-of-the-art intelligibility scores (CER/WER) despite being trained on significantly less data than competing models. While Koel-TTS outperforms LLM-based baselines (VALLE-X and XTTS-v2) in SSIM scores, it slightly underperforms Conditional Flow Matching (CFM)-based systems (F5-TTS and E2-TTS), which leverage $100k+$ hours of speech data, compared to $21k$ hours for our largest model. Human evaluations of naturalness (MOS) and speaker similarity (SMOS) show Koel-TTS to be equally or more preferred compared to all other models. We attribute the difference between SSIM scores and SMOS to SSIM's emphasis on timbre similarity, whereas human ratings consider additional factors such as style and accent. CMOS results in Figure 6, further confirm that Koel-TTS is preferred over all competing approaches.

Table 2: Intelligibility, SSIM and naturalness evaluation of various zero-shot TTS models on a subset of *test-clean* LibriTTS data.

Model	CER (%) ↓	WER (%) ↓	SSIM ↑	MOS ↑	SMOS ↑
Ground Truth	0.80	1.83	0.771	3.937 ± 0.028	-
VALLE-X (Zhang et al., 2023)	6.65	11.28	0.679	3.532 ± 0.046	3.709 ± 0.045
YourTTS (Casanova et al., 2022)	2.44	5.19	0.581	3.235 ± 0.047	3.229 ± 0.053
T5-TTS (Neekhara et al., 2024a)	1.66	3.28	0.459	3.533 ± 0.046	3.366 ± 0.050
E2-TTS (Eskimez et al., 2024)	1.29	2.66	**0.848**	3.889 ± 0.040	3.793 ± 0.045
F5-TTS (Chen et al., 2024)	1.23	2.55	0.834	3.930 ± 0.042	3.785 ± 0.045
XTTS-v2 (Casanova et al., 2024)	0.99	2.09	0.680	3.715 ± 0.043	3.434 ± 0.050
StyleTTS-2 (Li et al., 2024)	0.75	1.52	0.579	4.047 ± 0.039	3.786 ± 0.044
Koel-TTS 380m English	**0.55**	**1.41**	0.726	4.054 ± 0.039	3.826 ± 0.044
Koel-TTS 1.1b Multilingual	0.63	1.42	0.740	$\mathbf{4.058 \pm 0.040}$	$\mathbf{3.848 \pm 0.043}$

4.2 Related Work

For text-generation, preference alignment techniques (Christiano et al., 2017; Ouyang et al., 2022; Shao et al., 2024; Rafailov et al., 2024; Adler et al., 2024) have been fundamental in improving usability and reasoning abilities of text LLMs. These approaches, including RLHF and offline ranking methods, are now being extended to speech and audio (Ouyang et al., 2022; Rafailov et al., 2024; Cideron et al., 2024). For instance, SpeechAlign (Zhang et al., 2024), proposes an iterative strategy to align speech language models with human preferences by addressing the distribution gap between golden AR tokens (from real speech) and synthetic AR tokens (generated during inference). Although ground truth speech can be used to guide preference optimization training, we show in Appendix C that it introduces inconsistencies due to its fundamentally different distribution from model-generated tokens. This issue makes preference-based optimization such as DPO less effective.

Previous works such as Seed-TTS (Anastassiou et al., 2024) and (Tian et al., 2025) explore preference alignment in TTS but fall short in design clarity and improvements over the baseline model. For example, Seed-TTS (Anastassiou et al., 2024) performs DPO but does not specify how the generations are ranked to create the preference pairs. Moreover, their gains are marginal as compared to the improvements we observe in our experiments. In Seed-TTS, the WER over the baseline model improved by 15% and SSIM improved by 0.5% on English TTS. In contrast, our models with only DPO finetuning achieve a WER reduction from 4.02% to 1.90% (67% improvement) and SSIM improvement from 0.637 to 0.667 (6.4% improvement) for English TTS. Similarly, (Tian et al., 2025) relies primarily on SSIM to construct chosen-rejected pairs, claiming that WER-based optimization is less effective for preference alignment, as it focuses more on local transcription errors. In contrast, we demonstrate that both CER and SSIM can be used as reward signals in the Pareto ranking procedure of the generations, yielding consistent improvements across intelligibility, speaker similarity, and naturalness.

While CFG has been successfully used in diffusion and flow-based speech generation models (Ho and Salimans, 2021; Le et al., 2024; Du et al., 2024; Chen et al., 2024; Eskimez et al., 2024), there has been limited application of CFG in pure autoregressive LLMs trained with the next token prediction loss. In our work, we demonstrate that by dropping out the conditioning inputs during training, we can effectively interpolate between conditional and unconditional logits during inference, significantly enhancing contextual adherence in Speech LLMs. Notably, we observe the complementary benefits of preference alignment and CFG demonstrating the best results are achieved when both the techniques are applied together.

5 Conclusion

We introduce Koel-TTS, an LLM-based TTS model that accurately and efficiently maps text and reference audio to acoustic speech tokens, achieving state-of-the-art zero-shot performance. We provide a principled approach for ranking generated outputs such that we can maximize gains from preference alignment training. By integrating preference alignment—guided by transcription accuracy and speaker similarity—and Classifier-Free Guidance, Koel-TTS significantly reduces hallucinations while enhancing intelligibility, speaker similarity, and naturalness of generated speech.

Limitations

In our work, we demonstrate that preference optimization can be effective in enhancing measurable attributes of the generated speech such as transcription accuracy and speaker similarity. However, speech has several other attributes that are hard to automatically measure such as prosody, naturalness, accent and artifacts. We recommend future work in designing reliable metrics that can be automatically computed and integrated into a controllable reward system suitable for preference optimization. Such a controllable reward system can be used for improving the desired aspects of generated speech.

DPO and RPO provide a scalable and stable framework for aligning generative models using precomputed pairwise comparisons. However, offline preference optimization methods lack a feedback loop between training and preference pair generation, meaning that the chosen-rejected pairs are not strictly from the output distribution of the current state of the model. The effectiveness of such techniques could be further improved with online preference optimization methods, where both the model updates and reward assignments are performed iteratively. Exploring such approaches for generative speech LLMs remains a promising direction for future work.

References

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, and 1 others. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*.

Rohan Badlani, Adrian Łańcucki, Kevin J. Shih, Rafael Valle, Wei Ping, and Bryan Catanzaro. 2022. One tts alignment to rule them all. In *ICASSP*.

Evelina Bakhturina, Vitaly Lavrukhin, Boris Ginsburg, and Yang Zhang. 2021. Hi-Fi Multi-Speaker English TTS Dataset. In *INTERSPEECH*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and 1 others. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. *INTERSPEECH*.

Edresson Casanova, Ryan Langman, Paarth Neekhara, Shehzeen Hussain, Jason Li, Subhankar Ghosh, Ante Jukić, and Sang-gil Lee. 2025. Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference. *ICASSP*.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*. PMLR.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*.

Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian Mcwilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, Matthieu Geist, Leonard Hussenot, Neil Zeghidour, and Andrea Agostinelli. 2024. MusicRL: Aligning music generation to human preferences. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR.

Jordan Darefsky, Ge Zhu, and Zhiyao Duan. 2024. Parakeet.

Kalyanmoy Deb. 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, and 1 others. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Shehzeen Hussain, Paarth Neekhara, Jocelyn Huang, Jason Li, and Boris Ginsburg. 2023. Ace-vc: Adaptive and controllable voice conversion using explicitly disentangled self-supervised speech representations. In *ICASSP*.

Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. 2022a. Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context. In *ICASSP*. IEEE.

Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. 2022b. Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Anurag Kumar, Ke Tan, Zhaoheng Ni, Pranay Manocha, Xiaohui Zhang, Ethan Henderson, and Buye Xu. 2023. Torchaudio-squim: Reference-less speech quality and intelligibility measures in torchaudio. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and 1 others. 2024. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in Neural Information Processing Systems*.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2024. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2024. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*.

Paarth Neekhara, Shehzeen Hussain, Subhankar Ghosh, Jason Li, Rafael Valle, Rohan Badlani, and Boris Ginsburg. 2024a. Improving robustness of llm-based speech synthesis by learning monotonic alignment. *INTERSPEECH*.

Paarth Neekhara, Shehzeen Samarah Hussain, Rafael Valle, Boris Ginsburg, Rishabh Ranjan, Shlomo Dubnov, Farinaz Koushanfar, and Julian Mcauley. 2024b. SelfVC: Voice conversion with iterative refinement using self transformations. In *Proceedings of the 41st International Conference on Machine Learning*.

Frederico S Oliveira, Edresson Casanova, Arnaldo Candido Junior, Anderson S Soares, and Arlindo R Galvão Filho. 2023. CML-TTS: A multilingual dataset for speech synthesis in low-resource languages. In *International Conference on Text, Speech, and Dialogue*, pages 188–199. Springer.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A Large-Scale Multilingual Dataset for Speech Research. In *INTERSPEECH*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. *Findings of the Association for Computational Linguistics: EMNLP*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. Ella-v: Stable neural codec language modeling with alignment-guided sequence reordering. *arXiv preprint arXiv:2401.07333*.

Onkar Susladkar, Vishesh Tripathi, and Biddwan Ahmed. 2024. Bahasa harmony: A comprehensive dataset for bahasa text-to-speech synthesis with discrete codec modeling of engen-tts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Jinchuan Tian, Chunlei Zhang, Jiatong Shi, Hao Zhang, Jianwei Yu, Shinji Watanabe, and Dong Yu. 2025. Preference alignment improves language model-based tts. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv:2301.02111*.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. 2024. SpeechX: Neural codec language model as a versatile speech transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Hainan Xu, Fei Jia, Somshubra Majumdar, He Huang, Shinji Watanabe, and Boris Ginsburg. 2023. Efficient sequence transduction by jointly predicting tokens and durations. In *International Conference on Machine Learning*. PMLR.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Haohan Guo, Xuankai Chang, Jiatong Shi, Jiang Bian, Zhou Zhao, and 1 others. 2024. Uniaudio: Towards universal audio generation with large language models. In *Forty-first International Conference on Machine Learning*.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. LibriTTS: A corpus derived from librispeech for text-to-speech. *INTERSPEECH*.

Dong Zhang, Zhaowei Li, Shimin Li, Xin Zhang, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2024. Speechalign: Aligning speech generation to human preferences. In *Adances in Neural Information Processing Systems*.

Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv:2303.03926*.

A Alternate Model Architectures

In addition to the decoder context Koel-TTS architecture described in our paper, we explore two alternate Koel-TTS architectures (See Figure 7):

SV Conditioned Koel-TTS: In this configuration, a speaker embedding vector is extracted from the context audio using a pre-trained SV model (Koluguri et al., 2022b). This embedding vector is projected to hidden dimension of the transformer network, temporally expanded (repeated across the time axis) and added to the text encoder's output. The resulting combined representation serves as the input to the cross-attention layers of the AR decoder, enabling the prediction of audio codes while conditioning on the speaker identity. The advantage of this design is the ability to leverage transfer learning from the SV model, thereby enhancing generalization in scenarios with limited data. However, since the speaker vector is a compressed representation that primarily preserves voice identity, it does not capture other nuanced aspects of the context audio, such as speaking style and accent, which limits control over the generated speech.

Multi encoder Koel-TTS: In this architecture, context audio tokens are processed by a dedicated context encoder, which is a separate NAR transformer encoder. The outputs of the context encoder and text encoder are fed into alternate cross-attention layers of the AR decoder, as illustrated in Figure 7c. This design allows for a clear separation of modalities, where each encoder operates independently, and the decoder employs dedicated cross-attention mechanisms to integrate the outputs. This model also allows cross-attention biasing over the text tokens independently for learning monotonic alignment, while allowing variable length context audios.

Table 3 presents the baseline results of different Koel-TTS architectures on seen and unseen English speakers, **without** incorporating preference alignment training or CFG inference. All three architectures achieve similar intelligibility, but the *decoder context* model outperforms the *multi encoder* model on unseen speaker similarity, while the latter performs slightly better on seen speakers. These results suggest that *decoder context* model generalizes better to unseen speakers making it a more suitable choice for zero-shot TTS. The *multi encoder* architecture tends to overfit to the training speakers, as indicated by worse speaker similarity (SSIM) on unseen speakers, and better speaker similarity on seen speakers across all our experiments (also in Table 4). While *SV conditioned* model also achieves similar SSIM as decoder context, perceptually, we find the decoder context model captures the intended style of the context audio better.

Table 3: Baseline TTS results on seen and unseen speakers for different Koel-TTS models, **without using CFG or preference alignment**. Lower CER(%) & WER(%) indicate higher intelligibility. Higher SSIM indicates higher speaker similarity to ground-truth.

Eval Set	Model	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS ↑
Seen Speakers	Ground Truth	0.51 ± 0.00	1.42 ± 0.00	0.763 ± 0.000	4.616 ± 0.03
	Decoder context	1.73 ± 0.60	2.98 ± 0.59	0.700 ± 0.001	4.350 ± 0.038
	SV Conditioned	$\mathbf{1.71 \pm 0.41}$	$\mathbf{2.82 \pm 0.41}$	0.697 ± 0.003	$\mathbf{4.360 \pm 0.021}$
	Multi Encoder	1.92 ± 0.68	3.02 ± 0.76	$\mathbf{0.712 \pm 0.002}$	4.346 ± 0.028
Unseen Speakers	Ground Truth	0.80 ± 0.00	1.83 ± 0.00	0.771 ± 0.000	4.588 ± 0.020
	Decoder Context	2.68 ± 1.13	4.02 ± 1.12	$\mathbf{0.637 \pm 0.008}$	$\mathbf{4.347 \pm 0.024}$
	SV Conditioned	3.12 ± 0.98	4.22 ± 1.02	0.619 ± 0.003	4.318 ± 0.034
	Multi Encoder	$\mathbf{2.56 \pm 1.44}$	$\mathbf{3.74 \pm 1.36}$	0.601 ± 0.004	4.318 ± 0.059

B DPO and RPO on all model architectures

We perform DPO and RPO preference optimization on all models and evaluate the preference aligned checkpoint with and without CFG. Results are reported in Table 4. We observe a significant improvement in CER, WER and SSIM across all baseline models, when preference alignment or CFG is done in isolation. Across most architectures, the best metrics are achieved by CFG inference on a preference aligned checkpoint (DPO + CFG or RPO + CFG). Both DPO and RPO perform similarly, but in practice, we find DPO to be more sensitive to β hyperparameter as compared to RPO.

C Comparison with SpeechAlign

To compare with a prior technique proposed in SpeechAlign (Zhang et al., 2024), we perform an ablation in which the ground-truth audio tokens are selected as our chosen examples (as opposed to generated samples). We use a subset of *train-clean-360* LibriTTS data as chosen examples, and the worst ranked of the 6 generations (for an input) as the rejected example, creating a similar sized preference dataset as our other experiments. We find that preference alignment algorithms find it trivial to differentiate ground-truth examples from generated ones with the preference loss reducing to nearly zero within the first few hundred iterations. With our default DPO hyperparameters (β=0.01, LR=2e-7) such a setup leads to model degeneration and a very high CER (>90%). Fine-tuning DPO hyperparameters (β=1.0, LR=1e-7) and early stopping, prevents model degeneration, but does not

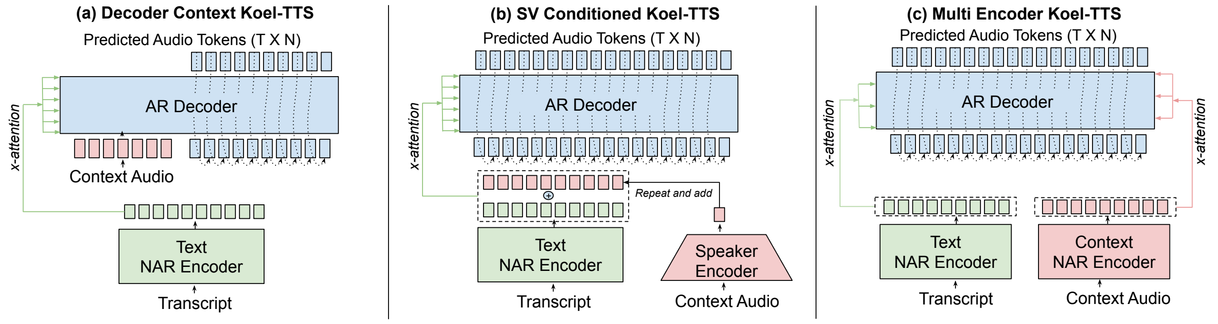


Figure 7: Koel-TTS Model Architectures: Three methods for conditioning TTS synthesis on context audio and transcripts. The *Decoder Context* approach utilizes the decoder's self-attention mechanism for speaker conditioning. The *Multi Encoder* and *SV Conditioned* models employ cross-attention layers for speaker conditioning.

Table 4: Evaluation of DPO, RPO and CFG on baseline models for all Koel-TTS architectures. We consider two DPO experiments with $\beta = (0.01, 0.05)$.

	Seen Speakers				*Unseen Speakers*			
Model/Technique	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS ↑	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS ↑
Multi Encoder (BL-1)	1.92 ± 0.68	3.02 ± 0.76	0.712 ± 0.002	4.346 ± 0.028	2.56 ± 1.44	3.74 ± 1.36	0.601 ± 0.004	4.318 ± 0.059
BL-1 + RPO ($\beta = 0.01$)	1.01 ± 0.58	1.76 ± 0.59	0.737 ± 0.002	4.408 ± 0.010	0.79 ± 0.12	1.72 ± 0.18	0.641 ± 0.002	4.389 ± 0.021
BL-1 + DPO ($\beta = 0.01$)	0.67 ± 0.17	1.48 ± 0.34	0.737 ± 0.004	4.406 ± 0.011	0.62 ± 0.12	1.49 ± 0.20	0.645 ± 0.001	4.402 ± 0.029
BL-1 + DPO ($\beta = 0.05$)	1.30 ± 0.51	2.25 ± 0.53	0.737 ± 0.003	4.401 ± 0.014	1.01 ± 0.40	2.01 ± 0.49	0.643 ± 0.005	4.402 ± 0.013
BL-1 + CFG ($\gamma = 2.5$)	0.73 ± 0.16	1.63 ± 0.24	0.752 ± 0.003	4.420 ± 0.010	0.69 ± 0.07	1.59 ± 0.10	0.653 ± 0.005	4.415 ± 0.007
BL-1 + RPO + CFG ($\gamma = 2.5$)	0.75 ± 0.23	1.56 ± 0.32	0.766 ± 0.003	$\mathbf{4.422 \pm 0.011}$	0.51 ± 0.12	1.25 ± 0.18	0.674 ± 0.004	4.392 ± 0.029
BL-1 + DPO ($\beta = 0.01$) + CFG ($\gamma = 2.5$)	$\mathbf{0.51 \pm 0.12}$	$\mathbf{1.32 \pm 0.23}$	$\mathbf{0.767 \pm 0.004}$	4.418 ± 0.012	0.58 ± 0.17	1.38 ± 0.08	$\mathbf{0.678 \pm 0.005}$	$\mathbf{4.417 \pm 0.015}$
BL-1 + DPO ($\beta = 0.05$) + CFG ($\gamma = 2.5$)	1.12 ± 0.83	1.87 ± 0.85	0.766 ± 0.002	4.420 ± 0.011	$\mathbf{0.49 \pm 0.07}$	$\mathbf{1.24 \pm 0.11}$	0.676 ± 0.004	4.390 ± 0.025
Decoder Context (BL-2)	1.73 ± 0.60	2.98 ± 0.59	0.700 ± 0.001	4.350 ± 0.038	2.68 ± 1.13	4.02 ± 1.12	0.637 ± 0.008	4.347 ± 0.024
BL-2 + RPO ($\beta = 0.01$)	1.01 ± 0.60	2.03 ± 0.62	0.719 ± 0.002	4.403 ± 0.013	1.17 ± 0.94	2.09 ± 1.00	0.681 ± 0.005	4.401 ± 0.012
BL-2 + DPO ($\beta = 0.01$)	1.32 ± 0.40	2.39 ± 0.46	0.708 ± 0.004	4.392 ± 0.017	0.89 ± 0.15	1.90 ± 0.28	0.667 ± 0.003	4.400 ± 0.012
BL-2 + DPO ($\beta = 0.05$)	1.25 ± 0.83	2.27 ± 0.97	0.716 ± 0.004	4.393 ± 0.016	0.98 ± 0.46	2.03 ± 0.49	0.676 ± 0.004	4.408 ± 0.010
BL-2 + CFG ($\gamma = 2.5$)	0.62 ± 0.20	1.58 ± 0.44	0.741 ± 0.003	$\mathbf{4.418 \pm 0.009}$	0.57 ± 0.11	$\mathbf{1.37 \pm 0.11}$	0.720 ± 0.004	$\mathbf{4.417 \pm 0.007}$
BL-2 + RPO + CFG ($\gamma = 2.5$)	$\mathbf{0.51 \pm 0.12}$	$\mathbf{1.38 \pm 0.25}$	$\mathbf{0.751 \pm 0.002}$	4.415 ± 0.013	$\mathbf{0.55 \pm 0.11}$	1.41 ± 0.19	$\mathbf{0.729 \pm 0.003}$	4.415 ± 0.012
BL-2 + DPO ($\beta = 0.01$) + CFG ($\gamma = 2.5$)	0.62 ± 0.09	1.53 ± 0.17	0.744 ± 0.002	4.409 ± 0.019	0.60 ± 0.10	1.40 ± 0.31	0.720 ± 0.001	4.387 ± 0.038
BL-2 + DPO ($\beta = 0.05$) + CFG ($\gamma = 2.5$)	0.54 ± 0.08	1.43 ± 0.19	0.749 ± 0.005	4.413 ± 0.018	0.55 ± 0.10	1.42 ± 0.28	$\mathbf{0.729 \pm 0.003}$	4.413 ± 0.013
SV Conditioned (BL-3)	1.71 ± 0.41	2.82 ± 0.41	0.697 ± 0.003	4.360 ± 0.021	3.12 ± 0.98	4.22 ± 1.02	0.619 ± 0.003	4.318 ± 0.034
BL-3 + RPO ($\beta = 0.01$)	0.72 ± 0.14	1.61 ± 0.22	0.717 ± 0.001	4.408 ± 0.010	1.25 ± 0.38	2.06 ± 0.49	0.668 ± 0.003	4.389 ± 0.026
BL-3 + DPO ($\beta = 0.01$)	0.62 ± 0.19	1.46 ± 0.31	0.705 ± 0.003	4.402 ± 0.011	0.76 ± 0.12	1.67 ± 0.11	0.650 ± 0.003	4.384 ± 0.023
BL-3 + DPO ($\beta = 0.05$)	1.24 ± 0.33	2.19 ± 0.33	0.713 ± 0.002	4.416 ± 0.042	1.64 ± 0.70	2.64 ± 0.69	0.663 ± 0.003	4.385 ± 0.019
BL-3 + CFG ($\gamma = 2.5$)	0.48 ± 0.12	1.38 ± 0.33	0.738 ± 0.003	4.407 ± 0.025	0.52 ± 0.11	1.38 ± 0.20	0.703 ± 0.002	$\mathbf{4.418 \pm 0.011}$
BL-3 + RPO + CFG ($\gamma = 2.5$)	0.46 ± 0.10	1.25 ± 0.14	$\mathbf{0.750 \pm 0.003}$	$\mathbf{4.423 \pm 0.010}$	0.57 ± 0.15	1.33 ± 0.21	$\mathbf{0.715 \pm 0.003}$	4.416 ± 0.014
BL-3 + DPO ($\beta = 0.01$) + CFG ($\gamma = 2.5$)	0.46 ± 0.05	$\mathbf{1.24 \pm 0.11}$	0.743 ± 0.002	4.417 ± 0.015	$\mathbf{0.47 \pm 0.07}$	$\mathbf{1.26 \pm 0.09}$	0.706 ± 0.004	4.412 ± 0.014
BL-3 + DPO ($\beta = 0.05$) + CFG ($\gamma = 2.5$)	$\mathbf{0.45 \pm 0.13}$	1.27 ± 0.06	0.747 ± 0.001	4.403 ± 0.021	0.70 ± 0.45	1.51 ± 0.44	$\mathbf{0.715 \pm 0.004}$	4.373 ± 0.055

yield significant improvement over the baseline model (Table 5). This suggests it is important to obtain chosen-rejected pairs from model's output distribution, for preference optimization such as DPO to work effectively.

Table 5: Comparison between preference alignment using generated output ranking (BL-1 + DPO), vs using GT audio tokens as chosen outputs in the preference pairs (BL-1 + DPO (GT as Chosen))

	Seen Speakers			
Model/Technique	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS ↑
Ground Truth	0.51 ± 0.00	1.42 ± 0.00	0.763 ± 0.000	4.616 ± 0.03
Multi Encoder (BL-1)	1.92 ± 0.68	3.02 ± 0.76	0.712 ± 0.002	4.346 ± 0.028
BL-1 + DPO	$\mathbf{0.67 \pm 0.17}$	$\mathbf{1.48 \pm 0.34}$	$\mathbf{0.737 \pm 0.004}$	$\mathbf{4.406 \pm 0.011}$
BL-1 + DPO (GT as Chosen)	1.58 ± 0.42	2.88 ± 0.41	0.710 ± 0.005	4.344 ± 0.038

D Pareto optimal ranking for creating preference pairs

Pareto optimal ranking is a technique for multi-attribute decision making (Deb, 2011). The key idea is to find non-dominated solutions and removing them from the current set recursively till we have ranked all items. When we find multiple items in the same pareto front, we break the ties by prioritizing our preference for more robust examples (lower CER), and we break any remaining ties by preferring higher SSIM. We provide the python code for ranking for this procedure in Listing 1.

E Multilingual Tokenization Ablations

We train three decoder-context Koel-TTS models considering three tokenization schemes besides phonemes — Model A: Aggregated characters from different languages (Vocab size = 256 × Number of Languages). Model A is the default model in our primary multilingual experiments. Model B: Shared character tokenizer (256 character tokens shared across all languages). Model C:

```
def pareto_ranking(items):
    """
    Given a list of (cer, ssim, item_idx), return the list of items
    sorted by their Pareto rank (rank 1 is best). Items in the same
    rank are sorted by ascending cer and incase of a tie, by descending ssim.

    :param items: List of tuples (cer, ssim, item_idx).
    :return: A list of tuples (rank, cer, ssim, item_idx), sorted first by rank,
             then by ascending cer within the same rank.
    """

    # A helper function to check if item A is dominated by item B
    # A: (cerA, ssimA, item_idxA), B: (cerB, ssimB, item_idxB)
    def is_dominated(A, B):
        return (B[0] <= A[0]) and (B[1] >= A[1]) and (B[2] != A[2])

    remaining = items[:]

    ranked_items = []  # Will hold tuples of (rank, cer, ssim, item_idx)
    current_rank = 1

    while remaining:
        # Find all non-dominated items in the current set 'remaining'
        non_dominated = []
        for i in range(len(remaining)):
            dominated = False
            for j in range(len(remaining)):
                if i != j:
                    if is_dominated(remaining[i], remaining[j]):
                        dominated = True
                        break
            if not dominated:
                non_dominated.append(remaining[i])

        # Assign current_rank to all non-dominated items
        # and remove them from remaining
        for nd in non_dominated:
            ranked_items.append((current_rank, nd[0], nd[1], nd[2]))
            remaining.remove(nd)

        current_rank += 1

    # Now sort the ranked items by (rank asc, cer asc, ssim desc)
    ranked_items.sort(key=lambda x: (x[0], x[1], -x[2]))

    return ranked_items
```

Listing 1: Pareto Optimal Ranking of generated outputs for a given text-context pair using CER and SSIM metrics

Multilingual sentence piece tokens [4] (Vocab size 110k). In all models, the we use separate phoneme tokens for each language and aggregate them into our tokenizer. We find that character-based tokenizers perform significantly better than sentence piece tokenizer on intelligibility metrics, especially when unseen words are encountered during inference. Table 6 compares the different models for each language studied in our work. All results are reported using CFG scale $\gamma = 2.5$, without any preference alignment. Additionally, we find the aggregated char tokenizer to perform better for cross-lingual TTS synthesis (when the context audio has a different language than the input text). This is because token embeddings for each language are independent from the others and not shared (as in the case of the shared character tokenizer).

F Evaluation on hard sentences with repeated words

Autoregressive TTS models often struggle with challenging sentences containing repeated words. Issues such as infinite silences, looping of words become more prominent when presented with such challenging inputs. While cross-attention biasing (Badlani et al., 2022; Neekhara et al., 2024a)

[4] https://huggingface.co/google-bert/bert-base-multilingual-uncased

Table 6: Comparison of decoder context Koel-TTS models trained using different text tokenizers, considering all allowed tokenizations at test time. (CFG Scale $\gamma = 2.5$, No Preference Alignment). Evaluation conduced on unseen speakers for each language on the test set described in Section 3.4

Model	Language	Tokenizer	CER(%) ↓	WER(%) ↓	SSIM ↑
Model A (Phoneme + Aggregated characters)	English	Phonemes	0.78 ± 0.77	1.65 ± 0.85	0.735 ± 0.002
Model B (Phoneme + Multilingual sentencepiece)	English	Phonemes	0.59 ± 0.10	1.57 ± 0.21	0.746 ± 0.003
Model C (Phoneme + Shared characters)	English	Phonemes	0.60 ± 0.20	1.41 ± 0.23	0.739 ± 0.002
Model B (Phoneme + Multilingual sentencepiece)	English	Multilingual sentencepiece	0.58 ± 0.09	1.44 ± 0.17	$\mathbf{0.747 \pm 0.000}$
Model C (Phoneme + Shared characters)	English	Shared characters	$\mathbf{0.52 \pm 0.04}$	$\mathbf{1.37 \pm 0.08}$	0.739 ± 0.001
Model A (Phoneme + Aggregated characters)	English	Aggregated characters	0.77 ± 0.49	1.67 ± 0.47	0.731 ± 0.005
Model A (Phoneme + Aggregated characters)	Spanish	Phonemes	1.37 ± 1.53	3.63 ± 1.99	0.693 ± 0.013
Model B (Phoneme + Multilingual sentencepiece)	Spanish	Phonemes	1.52 ± 0.71	4.01 ± 0.35	0.698 ± 0.011
Model C (Phoneme + Shared characters)	Spanish	Phonemes	1.41 ± 0.56	3.73 ± 0.59	0.703 ± 0.007
Model B (Phoneme + Multilingual sentencepiece)	Spanish	Multilingual sentencepiece	1.96 ± 0.68	4.93 ± 0.67	0.701 ± 0.007
Model C (Phoneme + Shared characters)	Spanish	Shared characters	1.64 ± 0.81	4.03 ± 1.08	$\mathbf{0.704 \pm 0.011}$
Model A (Phoneme + Aggregated characters)	Spanish	Aggregated characters	$\mathbf{1.03 \pm 0.11}$	$\mathbf{3.11 \pm 0.18}$	0.693 ± 0.008
Model A (Phoneme + Aggregated characters)	German	Phonemes	2.69 ± 2.36	5.75 ± 3.25	0.607 ± 0.008
Model B (Phoneme + Multilingual sentencepiece)	German	Phonemes	4.01 ± 3.89	6.88 ± 5.05	0.613 ± 0.021
Model C (Phoneme + Shared characters)	German	Phonemes	$\mathbf{1.67 \pm 0.71}$	$\mathbf{4.54 \pm 1.05}$	$\mathbf{0.645 \pm 0.007}$
Model B (Phoneme + Multilingual sentencepiece)	German	Multilingual sentencepiece	4.28 ± 3.13	7.77 ± 2.90	0.611 ± 0.014
Model C (Phoneme + Shared characters)	German	Shared characters	1.93 ± 1.06	4.76 ± 1.08	0.644 ± 0.005
Model A (Phoneme + Aggregated characters)	German	Aggregated characters	2.02 ± 1.13	4.81 ± 0.95	0.614 ± 0.003
Model B (Phoneme + Multilingual sentencepiece)	French	Multilingual sentencepiece	6.77 ± 2.12	10.33 ± 2.17	0.638 ± 0.009
Model C (Phoneme + Shared characters)	French	Shared characters	2.30 ± 0.70	$\mathbf{5.35 \pm 0.81}$	$\mathbf{0.643 \pm 0.006}$
Model A (Phoneme + Aggregated characters)	French	Aggregated characters	$\mathbf{2.28 \pm 1.26}$	5.50 ± 1.57	0.641 ± 0.011
Model B (Phoneme + Multilingual sentencepiece)	Italian	Multilingual sentencepiece	3.68 ± 0.65	12.83 ± 1.07	$\mathbf{0.650 \pm 0.003}$
Model C (Phoneme + Shared characters)	Italian	Shared characters	4.99 ± 1.83	12.81 ± 2.15	0.649 ± 0.009
Model A (Phoneme + Aggregated characters)	Italian	Aggregated characters	$\mathbf{2.37 \pm 0.50}$	$\mathbf{9.64 \pm 1.09}$	0.615 ± 0.003
Model B (Phoneme + Multilingual sentencepiece)	Dutch	Multilingual sentencepiece	4.19 ± 1.24	11.49 ± 1.96	0.607 ± 0.004
Model C (Phoneme + Shared characters)	Dutch	Shared characters	$\mathbf{3.05 \pm 0.93}$	$\mathbf{9.79 \pm 1.57}$	$\mathbf{0.613 \pm 0.008}$
Model A (Phoneme + Aggregated characters)	Dutch	Aggregated characters	3.32 ± 1.51	10.14 ± 1.18	0.594 ± 0.006

Table 7: Intelligibility and Speaker similarity evaluation on challenging sentences with repeated words on base models and the preference aligned models with CFG.

Model	CER(%) ↓	WER(%) ↓	SSIM ↑
Multi Encoder (Baseline)	5.62 ± 0.52	10.60 ± 0.90	0.788 ± 0.002
Multi Encoder (w Pref Align and CFG)	$\mathbf{5.03 \pm 0.16}$	$\mathbf{9.19 \pm 0.46}$	$\mathbf{0.807 \pm 0.001}$
Decoder Context (Baseline)	5.70 ± 0.56	10.38 ± 0.35	0.790 ± 0.003
Decoder Context (w Pref Align and CFG)	$\mathbf{4.69 \pm 0.14}$	$\mathbf{9.04 \pm 0.24}$	$\mathbf{0.798 \pm 0.002}$
SV Conditioned (Baseline)	5.59 ± 0.84	10.33 ± 0.94	0.785 ± 0.001
SV Conditioned (w Pref Align and CFG)	$\mathbf{4.67 \pm 0.37}$	$\mathbf{8.50 \pm 0.66}$	$\mathbf{0.798 \pm 0.002}$

partially addresses this issue, we find CFG and preference alignment can further improve robustness of the base model by mitigating hallucinations. Table 7 presents evaluation of base model and preference aligned + CFG models on a set of 91 hard sentences linked in our webpage. We conduct these evaluations by pairing the challenging texts with 2 seen speakers in our training dataset. As indicated by the results, while CFG and preference alignment improve CER and WER, there is scope for further improvement using inference-time monotonic alignment strategies.

G SSIM with WavLM

We provide additional SSIM evaluations using a different speaker verification model architecture WavLM (Chen et al., 2022) in Table 8. We observe similar improvements with CFG and preference alignment as reported by Titanet small in Table 1

Table 8: Preference Alignment (DPO, RPO) and CFG improvements over a baseline *Koel-TTS 380m English* model for zero-shot TTS using WavLM as the speaker verification model.

Model/Technique	CER(%) ↓	WER(%) ↓	SSIM ↑	Squim-MOS
Ground Truth	0.80	1.83	0.771	3.93 ± 0.03
Baseline (BL)	2.68 ± 1.13	4.02 ± 1.12	0.929	4.35 ± 0.02
BL + DPO	0.89 ± 0.15	1.90 ± 0.28	0.940	4.40 ± 0.01
BL + RPO	1.17 ± 0.94	2.09 ± 1.00	0.943	4.40 ± 0.01
BL + CFG	0.57 ± 0.11	$\mathbf{1.37 \pm 0.11}$	0.944	$\mathbf{4.42 \pm 0.01}$
BL + DPO + CFG	$\mathbf{0.55 \pm 0.10}$	1.42 ± 0.28	0.945	4.41 ± 0.01
BL + RPO + CFG	$\mathbf{0.55 \pm 0.11}$	1.41 ± 0.19	0.946	$\mathbf{4.42 \pm 0.01}$

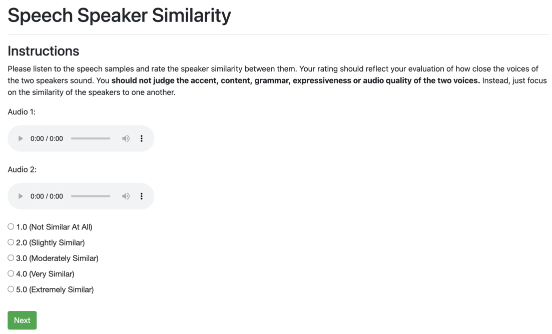
H MOS, SMOS and CMOS Evaluation

Naturalness MOS Evaluation: We ask human listeners to rate the audio on a scale of 1 to 5 point naturalness scale with 1 point increments. We present 180 audio examples of each technique and each audio is independently rated by at least 11 listeners. This results in a total of 1980 evaluations per technique. The template used for the Naturalness human study is shown in Figure 10. We report the

MOS with 95% confidence intervals in Table 2 of the paper.

Speaker Similarity MOS (SMOS): For SMOS evaluation, we ask human listeners to rate the speaker similarity of a given pair of utterances. For this evaluation, each synthetic utterance is paired with a real context utterance of the target speaker. We create pairs for all of the 180 synthesized utterances of each technique. Each pair is rated by at least 11 independent listeners resulting in at least 1800 speaker similarity evaluations of each technique. We ask the listeners to judge only the voice/speaker of the utterances and ignore the accent, content, grammar and expressiveness of speech. following past work (Casanova et al., 2022; Hussain et al., 2023; Neekhara et al., 2024b). The templates used for this user study are shown in Figures 8, 9 and 10.

Comparative MOS (CMOS): For CMOS, listeners are asked to compare two audio utterances on naturalness and indicate their preference as one of the five options shown in Figure 9. We pair the two Koel-TTS models with all other models. We evaluate the percentage of times across 1800 evaluations that Koel-TTS is preferred over an alternate model.


Speech Speaker Similarity

Instructions

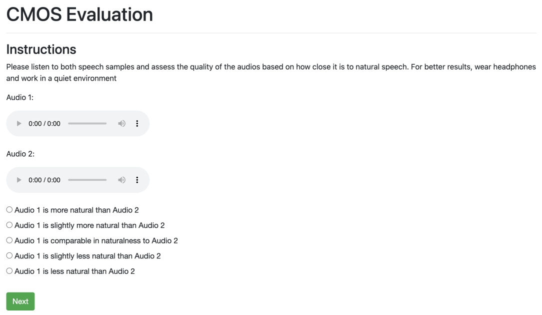
Please listen to the speech samples and rate the speaker similarity between them. Your rating should reflect your evaluation of how close the voices of the two speakers sound. You **should not judge the accent, content, grammar, expressiveness or audio quality of the two voices.** Instead, just focus on the similarity of the speakers to one another.

Audio 1:

Audio 2:

○ 1.0 (Not Similar At All)
○ 2.0 (Slightly Similar)
○ 3.0 (Moderately Similar)
○ 4.0 (Very Similar)
○ 5.0 (Extremely Similar)

Figure 8: User Study template used for Speaker Similarity (SMOS) evaluation


CMOS Evaluation

Instructions

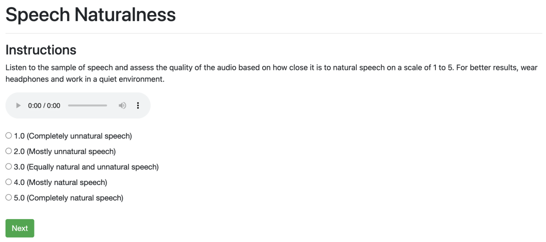
Please listen to both speech samples and assess the quality of the audios based on how close it is to natural speech. For better results, wear headphones and work in a quiet environment

Audio 1:

Audio 2:

○ Audio 1 is more natural than Audio 2
○ Audio 1 is slightly more natural than Audio 2
○ Audio 1 is comparable in naturalness to Audio 2
○ Audio 1 is slightly less natural than Audio 2
○ Audio 1 is less natural than Audio 2

Figure 9: User Study template used for Comparative-CMOS evaluation


Speech Naturalness

Instructions

Listen to the sample of speech and assess the quality of the audio based on how close it is to natural speech on a scale of 1 to 5. For better results, wear headphones and work in a quiet environment.

○ 1.0 (Completely unnatural speech)
○ 2.0 (Mostly unnatural speech)
○ 3.0 (Equally natural and unnatural speech)
○ 4.0 (Mostly natural speech)
○ 5.0 (Completely natural speech)

Figure 10: User Study template used for MOS evaluation